



[Subscribe](#) (Full Service) [Register](#) (Limited Service, Free) [Login](#)

Search: ☒ The ACM Digital Library ☐ The Guide

(hardware <near/3> bytecode AND software <near/3> byteco



[Feedback](#) [Report a problem](#) [Satisf](#)

Terms used

hardware near/3 bytecode AND software near/3 bytecode AND configur% near/6 hardware AND virtual near

Sort results by

Display results

[Save results to a Binder](#)

[Search Tips](#)

☐ Open results in a new window

Try an [Advanced Search](#)

Try this search in [The A](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

1 [Software support: VMSTAR: synthesizing scalable runtime environments for sensor netwo](#)



Joel Koshy, Raju Pandey

November 2005 **Proceedings of the 3rd international conference on Embedded networked s
SenSys '05**

Publisher: ACM Press

Full text available: pdf (159.40 KB)

Additional Information: [full citation](#), [absiract](#), [references](#), [index terms](#)

Sensor networks are being deployed at massive scales, containing a range of platforms. Program for sensor networks should meet the attendant challenges of scale and heterogeneity. Research considered virtual machines as a means to address these challenges. However, in order to satisfy limitations of sensor nodes, they export only a minimal set of services to the application program applications of even moderate complexity difficult to implement. ...

Keywords: network reprogramming, operating systems, programming languages, software syr machines, wireless sensor networks

2 [Virtual machine monitors: Xen and the art of virtualization](#)



Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Warfield

October 2003 **Proceedings of the nineteenth ACM symposium on Operating systems princip**

Publisher: ACM Press

Full text available: pdf (168.76 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citing](#), [index](#)

Numerous systems have been designed which use virtualization to subdivide the ample resource computer. Some require specialized hardware, or cannot support commodity operating systems 100% binary compatibility at the expense of performance. Others sacrifice security or functionality for resource isolation or performance guarantees; most provide only best-effort provisioning, service. This paper presents Xen, an x86 virtual machine monitor ...

Keywords: hypervisors, paravirtualization, virtual machine monitors

3 [Formalizing the safety of Java, the Java virtual machine, and Java card](#)



Pieter H. Hartel, Luc Moreau

December 2001 **ACM Computing Surveys (CSUR)**, Volume 33 Issue 4

Publisher: ACM Press

Full text available:  [pdf\(442.86 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index](#)

We review the existing literature on Java safety, emphasizing formal approaches, and the impact of small footprint devices such as smartcards. The conclusion is that although a lot of good work has been done, more concerted effort is needed to build a coherent set of machine-readable formal models of the language and its implementation. This is a formidable task but we believe it is essential to build trust in Java in order to achieve ITSEC level 6 or Common Criteria ...

Keywords: Common criteria, programming

4 [SableSpMT: a software framework for analysing speculative multithreading in Java](#)



Christopher J. F. Pickett, Clark Verbrugge

September 2005 **ACM SIGSOFT Software Engineering Notes , The 6th ACM SIGPLAN-SIGSOFT Program analysis for software tools and engineering PASTE '05**, Volume 31 Issue 5

Publisher: ACM Press

Full text available:  [pdf\(602.03 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#)

Speculative multithreading (SpMT) is a promising optimisation technique for achieving faster execution of sequential programs on multiprocessor hardware. Analysis of and data acquisition from such systems is difficult and complex, and is typically limited to a specific hardware design and simulation environment. We have implemented a flexible, software-based speculative multithreading architecture within the context of the featured Java virtual machine. We consider the entire Java language ...

Keywords: java, profiling, speculative multithreading, static and dynamic analysis, thread level virtual machines


5 [Design and implementation of a distributed virtual machine for networked computers](#)



Emin Gün Sirer, Robert Grimm, Arthur J. Gregory, Brian N. Bershad

December 1999 **ACM SIGOPS Operating Systems Review , Proceedings of the seventeenth ACM conference on Operating systems principles SOSP '99**, Volume 33 Issue 5

Publisher: ACM Press

Full text available:  [pdf\(1.62 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index](#)

This paper describes the motivation, architecture and performance of a distributed virtual machine for networked computers. DVMS rely on a distributed service architecture to meet the manageability and uniformity requirements of large, heterogeneous clusters of networked computers. In a DVM, system services such as verification, security enforcement, compilation and optimization, are factored out of client machines and run on powerful network servers. This partitioning of system functions ...

6 [How java programs interact with virtual machines at the microarchitectural level](#)



Lieven Eeckhout, Andy Georges, Koen De Bosschere

October 2003 **ACM SIGPLAN Notices , Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '03**, Volume 38 Issue 10

Publisher: ACM Press

Full text available:  [pdf\(348.88 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index](#)

Java workloads are becoming increasingly prominent on various platforms ranging from embedded systems to general-purpose computers to high-end servers. Understanding the implications of all the aspects of running Java workloads, is thus extremely important during the design of a system that will run Java. In other words, understanding the interaction between the Java application, its input and the virtual machine it runs on, is key to a successful design. The goal of this paper ...

Keywords: Java workloads, performance analysis, statistical data analysis, virtual machine technology characterization

- 7 Compilation and run-time systems: DELI: a new run-time control point
Giuseppe Desoli, Nikolay Mateev, Evelyn Duesterwald, Paolo Faraboschi, Joseph A. Fisher
November 2002 **Proceedings of the 35th annual ACM/IEEE international symposium on Microarchitecture**
Publisher: IEEE Computer Society Press


Full text available:  [pdf\(1.27 MB\)](#)  [Publisher Site](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The Dynamic Execution Layer Interface (DELI) offers the following unique capability: it provides over the execution of programs, by allowing its clients to observe and optionally manipulate every instruction---at run time---just before it runs. DELI accomplishes this by opening up an interface between the execution of software and hardware. To avoid the slowdown, DELI caches a private executed code and always runs out of its own private cache. In ...


- 8 LLVA: A Low-level Virtual Instruction Set Architecture
Vikram Adve, Chris Lattner, Michael Brukman, Anand Shukla, Brian Gaeke
December 2003 **Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture**
Publisher: IEEE Computer Society


Full text available:  [pdf\(196.08 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

A virtual instruction set architecture (V-ISA) implemented via a processor-specific software translator provide great flexibility to processor designers. Recent examples such as Crusoe and DAISY, however, existing hardware instruction sets as virtual ISAs, which complicates translation and optimization. There has been little research on specific designs for a virtual ISA for processors. This paper proposes a new (LLVA) and a translation strategy for implementing ...

- 9 Relational profiling: enabling thread-level parallelism in virtual machines
 Timothy Heil, James E. Smith
December 2000 **Proceedings of the 33rd annual ACM/IEEE international symposium on Microarchitecture**
Publisher: ACM Press


Full text available:  [pdf\(237.19 KB\)](#)  [ps\(1.61 MB\)](#)  [Publisher Site](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

- 10 Distributed VEEs: PDS: a virtual execution environment for software deployment
 Bowen Alpern, Joshua Auerbach, Vasanth Bala, Thomas Frauenhofer, Todd Mummert, Michael Pigo
June 2005 **Proceedings of the 1st ACM/USENIX international conference on Virtual execution environments**
Publisher: ACM Press

Full text available:  [pdf\(299.26 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The Progressive Deployment System (PDS) is a virtual execution environment and infrastructure specifically for deploying software, or "assets", on demand while enabling management from a remote host. PDS intercepts a select subset of system calls on the target machine to provide a partial virtualized operating system level. This enables an asset's install-time environment to be reproduced virtually, otherwise not isolating the asset from peer applications on the target ...


Keywords: deployment, installation, management, streaming, virtualization

- 11 A portable Virtual Machine target for Proof-Carrying Code
 Michael Franz, Deepak Chandra, Andreas Gal, Vivek Haldar, Fermín Reig, Ning Wang
June 2003 **Proceedings of the 2003 workshop on Interpreters, virtual machines and emulators**
Publisher: ACM Press

Full text available:  [pdf\(285.85 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

Virtual Machines (VMs) and Proof-Carrying Code (PCC) are two techniques that have been used to provide safety for (mobile) code. Existing virtual machines, such as the Java VM, have several drawbacks. First, the effort required for safety verification is considerable. Second and more subtly, the need to provide verification by the code consumer inhibits the amount of optimization that can be performed by the producer. This in turn makes just-in-time compilation surprising ...

12 Application isolation in the Java Virtual Machine

 Grzegorz Czajkowski
October 2000 **ACM SIGPLAN Notices , Proceedings of the 15th ACM SIGPLAN conference on programming, systems, languages, and applications OOPSLA '00**, Volume 35 Issue 10
Publisher: ACM Press

Full text available:  [pdf\(217.49 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index](#)

To date, systems offering multitasking for the Java[®] programming language either use or not use a class loader for each application. Both approaches are unsatisfactory. Using operating system protection is expensive, scales poorly and does not fully exploit the protection features inherent in a safe language. Class loaders replicate application code, obscure the type system, and non-uniformly treat 'trusted' and 'untrusted' classes, which leads to subtle, but nevertheless, potential security holes ...

Keywords: Java Virtual Machine, application isolation, multitasking

13 The Jrpm system for dynamically parallelizing Java programs


 Michael K. Chen, Kunle Olukotun
May 2003 **ACM SIGARCH Computer Architecture News , Proceedings of the 30th annual symposium on Computer architecture ISCA '03**, Volume 31 Issue 2
Publisher: ACM Press

Full text available:  [pdf\(320.42 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

We describe the Java runtime parallelizing machine (Jrpm), a complete system for parallelizing Java programs automatically. Jrpm is based on a chip multiprocessor (CMP) with thread-level speculation support. CMPs have low sharing and communication costs relative to traditional multiprocessors. Thread-level speculation (TLS) simplifies program parallelization by allowing us to parallelize optimistically without the need for correct sequential program behavior. Using a Java virtual machine ...

14 When to use a compilation service?

 Jeffrey Palm, Han Lee, Amer Diwan, J. Eliot B. Moss
June 2002 **ACM SIGPLAN Notices , Proceedings of the joint conference on Languages, compilers, and tools for embedded systems: software and compilers for embedded systems LCTES '02**, Volume 37 Issue 7
Publisher: ACM Press


Full text available:  [pdf\(365.49 KB\)](#)


Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Modern handheld computers are certainly capable of running general purpose applications, such as Java. However, short battery life rather than computational capability often limits the usefulness of these computers. This paper considers how to reduce the energy consumption of Java applications. There are three interleaved steps in running Java programs in a compiled environment: downloading bytecodes, compiling and possibly optimizing the bytecodes, and running ...

Keywords: Java, distributed compilation, energy efficient compilation


15 Session summaries from the 17th symposium on operating systems principles (SOSP'99)

 Jay Lepreau, Eric Eide
April 2000 **ACM SIGOPS Operating Systems Review**, Volume 34 Issue 2
Publisher: ACM Press

Full text available:  pdf(3.15 MB)

Additional Information: [full citation](#), [index terms](#)

16 Java driven codesign and prototyping of networked embedded systems

 Josef Fleischmann, Klaus Buchenrieder, Rainer Kress

June 1999 **Proceedings of the 36th ACM/IEEE conference on Design automation**

Publisher: ACM Press

Full text available:  pdf(69.24 KB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

17 Improving 64-Bit Java IPF Performance by Compressing Heap References

Ali-Reza Adl-Tabatabai, Jay Bharadwaj, Michal Cierniak, Marsha Eng, Jesse Fang, Brian T. Lewis, B James M. Stichnoth

March 2004 **Proceedings of the international symposium on Code generation and optimization directed and runtime optimization CGO '04**


Publisher: IEEE Computer Society

Full text available:  pdf(172.84 KB)

Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

64-bit processor architectures like the Intel® Itanium® Processor Family are designed for large need large memory addresses. When running applications that fit within a 32-bit address space, a disadvantage compared to 32-bit CPUs because of the larger memory footprints for their data, worse cache and TLB utilization, and consequently lower performance because of increased miss considers software techniques for virtual machines that all ...

18 Prototyping and validation techniques: Rappit: framework for synthesis of host-assisted software for adaptive embedded systems

 Jiwon Hahn, Qiang Xie, Pai H. Chou

September 2005 **Proceedings of the 3rd IEEE/ACM/IFIP international conference on Hardware codesign and system synthesis CODES+ISSS '05**

Publisher: ACM Press

Full text available:  pdf(1.27 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Scripting is a powerful, high-level, cross-platform, dynamic, easy way of composing software modules. Unfortunately, the high runtime overhead has prevented scripting from being widely adopted applications. This work proposes to overcome these obstacles by synthesizing light-weight, host engines for embedded systems. The result is dramatically shortened development cycle due to 1 level abstraction, interactive access and dynamic reconfig ...

Keywords: adaptive systems, scripting, software synthesis

19 A Model-Based Approach for Executable Specifications on Reconfigurable Hardware

Tim Schattkowsky, Wolfgang Mueller, Achim Rettberg

March 2005 **Proceedings of the conference on Design, Automation and Test in Europe - Vols. 1-4**

Publisher: IEEE Computer Society

Full text available:  pdf(174.47 KB)

Additional Information: [full citation](#), [abstract](#)

UML 2.0 provides a rich set of diagrams for systems documentation and specification. Many efforts undertaken to employ different aspects of UML for multiple domains, mainly in the area of software. Considering the area of electronic design automation, however, we currently see only very few . investigate UML for hardware design and hardware/software co-design. In this article, we present executable UML closing the gap from system specification to ...

20

A framework for reducing the cost of instrumented code



Matthew Arnold, Barbara G. Ryder

May 2001

**ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2001 conference on
language design and implementation PLDI '01**, Volume 36 Issue 5

Publisher: ACM Press

Full text available: pdf(1.78 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index](#)

Instrumenting code to collect profiling information can cause substantial execution overhead. This makes instrumentation difficult to perform at runtime, often preventing many known *offline* free optimizations from being used in online systems. This paper presents a general framework for *instrumentation sampling* to reduce the overhead of previously expensive instrumentation. The simple and effective, using code-duplication and *count* ...

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)